UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/841,503	04/24/2001	Richard Alan Dayan	RPS9 2001 0011	5669

53493 7590 12/13/2007

LENOVO (US) IP Law
1009 Think Place
Building One, 4th Floor 4B6
Morrisville, NC 27560

EXAMINER	
HENNING, MATTHEW T	

ART UNIT	PAPER NUMBER
2131	

MAIL DATE	DELIVERY MODE
12/13/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Application Number: 09/841,503
Filing Date: April 24, 2001
Appellant(s): DAYAN ET AL.

Ronald V. Davidge
Reg. No. 33,863
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed September 21st, 2007 appealing from the Office action mailed March 21st, 2007.

(1) Real Party in Interest

A statement identifying by name the real party in interest is contained in the brief.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The statement of the status of claims contained in the brief is incorrect. A correct statement of the status of the claims is as follows:

The amendment filed on September 12th, 2007 has been entered, and as such the following changes to the status of the claims have been made.

Claims 44 and 57 have been amended subsequent to the final rejection.

Claims 44-49 and 57-62 stand rejected as shown in the grounds of rejection below.

Claim 1-43, and 50-56 have been canceled.

(4) Status of Amendments After Final

The appellant's statement of the status of amendments after final rejection contained in the brief is incorrect.

The amendment after final rejection filed on September 12, 2007 has been entered.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

The examiner notes that the appellant has grouped claims 49 and 62 with claims 44 and 57, and has not argued these claims separately, although these claims were rejected further in view of SCHMIDT. The examiner assumes that the appellant wished for claims 49 and 62, to either fall or stand with claims 44 and 57, and as such has not addressed these claims separately.

(7) Claims Appendix

Due to the entry of the amendment filed on September 12th, 2007, the claims appendix is not correct. However, because the amendments were made to rewrite claims 44 and 57 into independent form by incorporating all the limitations of claims 37-38 into claim 44 as well as claims 50-51 into claim 57, the claims appendix has not been held improper.

(8) Evidence Relied Upon

6,026,016	GAFKEN	2-2000
5,128,995	ARNOLD et al.	7-1992
6,088,759	HASBUN et al.	7-2000
2001/0039651A1	HAYASHI et al.	11-2001
5,826,015	SCHMIDT	10-1998

A. Menezes et al. "Handbook of Applied Cryptography", CRC Press, 1997, pp. 397-405

(9) Grounds of Rejection

The following grounds of rejection remain the same as the grounds of rejection presented in the office action dated March 21st, 2007, but have been rewritten as a result of the amendment to the claims dated September 12th, 2007.

The following ground(s) of rejection are applicable to the appealed claims:

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

Claims 44-48 and 57-61 are rejected under 35 U.S.C. 103(a) as being unpatentable over Gafken (US Patent Number 6,026,016), further in view of Arnold et al. (US Patent Number 5,128,995) hereinafter referred to as Arnold, and further in view of Menezes et al. ("Handbook of Applied Cryptography") hereinafter referred to as Menezes, and further in view of Hasbun et al. (U.S. Patent Number 6,088,759) hereinafter referred to as Hasbun, and further in view of Hayashi et al. (US 2001/0039651 A1) hereinafter referred to as Hayashi.

Regarding claim 44, Gafken disclosed a method for providing a capability to securely update information stored in a plurality of computer systems (See Gafken Fig. 5), where in the method comprises: forming a protected partition within each of the computer systems (See Gafken Col. 4 Paragraphs 3-4); storing within nonvolatile storage (See Gafken Fig. 1 Element 118) of each computer system in the plurality of computer systems, an operating system (See Gafken Fig. 1 Element 150), and an initialization routine (See Gafken Fig. 1 Element 151) to execute within a processor of the computer system after power on of the computer system (See Gafken Col. 3 Paragraph 2 Lines 1-4), wherein the initialization routine includes instructions causing the protected partition to be locked before the operating system is loaded (See Gafken Col. 13 Paragraph 9 – Col. 14 Paragraph 2), and wherein instructions causing information stored within a predetermined location to be written within the protected partition after predetermined security procedures have occurred but before the protected partition is locked (See Gafken Col. 13 Paragraph 8); and wherein the initialization routine includes instructions causing the processor of the computer system to perform a method including: locking the protected partition to prevent further modification of information stored within the protected partition (See Gafken Col. 13 Paragraph 9 – Col. 14 Paragraph 1), establishing a network connecting each computer system in the plurality of computer systems with a server system (See Gafken Col. 3 Paragraph 6 and Col. 12 Paragraph 7); generating an update partition file within the server system (See Gafken Col. 12 Paragraph 7 – Col. 13 Paragraph 1); the update partition file includes an entry (See Gafken Col. 12 Lines 53-67 "code image") and a plurality of encrypted elements (See Gafken Col. 12 Lines 53-67 "BIOS Signature" wherein one interpretation of the claims each binary bit of the signature reads on one of the plurality of encrypted elements), each encrypted element within the plurality of encrypted elements is associated with the entry (See Gafken Col. 12 Line 53 - Col. 13 Line 2), transmitting the update partition file over the network to each computer system in the plurality of computer systems (See Gafken Col. 12 Paragraph 7); and following determining that the update partition file is stored within the computing system for updating the protected partition, verifying whether the entry within the update partition file has been generated by the server system (See Gafken Col. 12 Line 53 – Col. 13 Line 2); and storing the update partition file within the predetermined location of each computer system in the plurality of computer systems (See Gafken Col. 12 Paragraph 5), wherein the entry within the update partition file is written to the protected partition only following verification that the entry has been generated by the server system (See Gafken Col. 12 Line 53 – Col. 13 Line 2).

However, Gafken failed to disclose the protected partition being within a hard drive; or that a setup password stored in the nonvolatile storage for use in the predetermined security procedures; wherein the initialization routine includes instructions causing the processor of the computer system to perform a method including: comparing information stored in the protected partition with information from the update partition file stored within the predetermined location; or that when a portion of the information stored in the protected partition is found partition is found to match a portion of the information stored within the update partition file, overwriting the portion of the information stored in the protected partition with the portion of the information stored in the update partition file if space around the portion of the information stored in the protected partition is sufficient; or that when a portion of the information stored in the protected partition is not found to match a portion of the information stored within the update partition file, writing the portion of the information stored within the update partition file to append to the information stored in the protected partition if space within the protected partition is sufficient; or that the update partition file included a plurality of entries and that each entry within the plurality of entries includes information to be stored at a different location within the protected partition; or verifying whether each entry in the plurality of entries within the update partition file has been generated by the server system; or that each entry in the plurality of entries within the update partition file is written to the protected partition only following verification that the entry has been generated by the server system. Gafken further failed to disclose a plurality of encrypted elements (in a more specific interpretation of encrypted element than a bit of a signature).

Gafken did disclosed that "although the example…describes a flash memory used to store…a BIOS…other types of nonvolatile memories storing other types of information may be used" (See Gafken Col. 14 Paragraph 6).

Arnold teaches that a BIOS can be stored in a protected partition of a hard drive (See Arnold Col. 2 Line 63 – Col. 3 Line 12).

It would have been obvious to the ordinary person skilled in the art at the time of invention to employ the teachings of Arnold in the BIOS updating system of Gafken by storing the BIOS in a protected partition of a hard drive instead of flash memory. This would have been obvious because the ordinary person skilled in the art would have been motivated to provide a fast and efficient way to store BIOS code.

Menezes teaches that providing a sequence number (password), stored and updated at both a receiver and a sender, in a digital signature of the sender, protects the signature against replay attacks (See Menezes Page 399 Section (ii).

It would have been obvious to the ordinary person skilled in the art at the time of invention to employ the teachings of Menezes to the validation signatures of Gafken by providing a sequence number in the signature of the update image. This would have been obvious because the ordinary person skilled in the art would have been motivated to provide protection against illicitly signed updates.

Hasbun teaches that a BIOS update can include a plurality of entries (blocks) wherein each entry within the plurality of entries includes information to be stored at a different location within the BIOS (See Hasbun Col. 5 Paragraph 6- Col. 6 Paragraph 2); comparing information stored in the protected partition with information from the update partition file stored within the predetermined location and when a portion of the information stored in the protected partition is found partition is found to match a portion of the information stored within the update partition file, overwriting the portion of the information stored in the protected partition with the portion of the information stored in the update partition file if space around the portion of the information stored in the protected partition is sufficient (See Hasbun Col. 5 Paragraph 6 – Col. 6 Paragraph 2 and Col. 12 Line 59 – Col. 16 Line 27 wherein Hasbun teaches that a bios update can be allocated into virtual blocks so that the blocks can be updated individually without having to erase the entire memory first); when a portion of the information stored in the protected partition is not found to match a portion of the information stored within the update partition file, writing the portion of the information stored within the update partition file to append to the information stored in the protected partition if space within the protected partition is sufficient (See Hasbun Col. 7 Paragraph 2 wherein Hasbun teaches that new blocks should be allocated from existing free memory).

It would have been obvious to the ordinary person skilled in the art at the time of invention to employ the teachings of Hasbun to the bios updating system of Gafken, Arnold, and Menezes by updating each update block of the BIOS one at a time in the manner taught by Hasbun. This would have been obvious <u>because</u> the ordinary person skilled in the art would have been motivated to provide a safe method for updating a bios without risking loss of the entire bios in the event of a power failure.

Hayashi teaches a method for providing a variety of software safely by breaking the file into pieces and decrypting each piece separately (See Hayashi Page 1 Col. 2 Paragraphs 3-10).

It would have been obvious to the ordinary person skilled in the art at the time of invention to employ the teachings of Hayashi to the updating system of the combination of Gafken, Arnold, Menezes, and Hasbun by signing blocks of the file separately from the other blocks, thereby obtaining a signature (encrypted element) for each block (entry), and thereby allowing each signature to be verified separate from the other blocks. This would have been obvious because the ordinary person skilled in the art would have been motivated to provide users with customized software without imposing too much of a load on the provider.

Regarding claim 57, Gafken disclosed an interconnected system for providing updated information in a secure manner (See Gafken Abstract and Fig. 5), wherein the interconnected system comprises: a network (See Gafken Col. 3 Paragraph 6 and Col. 12 Paragraph 7); a server system connected to the network and programmed to generate an update partition file and to transmit the update partition file over the network (See Gafken Col. 12 Paragraph 7 – Col. 13 Paragraph 1); a computer system connected to the network, wherein the computer system includes a processor (See Gafken Fig. 1), non-volatile data storage including a protected partition (See Gafken Fig. 1 Element 115 and Col. 4 Paragraphs 3-4), wherein the processor is programmed to receive the update partition file from the network and to store the update partition file in a predetermined location within the nonvolatile data storage outside the protected partition (See Gafken Col. 12 Paragraphs 5-7), and wherein the nonvolatile data storage stores an operating system and an initialization routine executing within the processor after power on of the computer system (See Gafken Fig. 1 Element 118 and Col. 3 Paragraph 2 Lines 1-4), including instructions causing the protected partition to be locked before the operating system is loaded (See Gafken Col. 13 Paragraph 9 – Col. 14 Paragraph 2), and instructions causing information stored within the predetermined location to be written within the protected partition after predetermined security procedures have occurred but before the protected partition is locked (See Gafken Col. 13 Paragraph 8), wherein the initialization routine includes instructions causing the protected partition to be locked before the operating system is loaded (See Gafken Col. 13 Paragraph 9 – Col. 14 Paragraph 2), and wherein instructions causing information stored within a predetermined location to be written within the protected partition after predetermined security procedures have occurred but before the protected partition is locked (See Gafken Col. 13 Paragraph 8); and wherein the initialization routine includes instructions causing the processor of the computer system to perform a method including: locking the protected partition to prevent further modification of information stored within the protected partition (See Gafken Col. 13 Paragraph 9 – Col. 14 Paragraph 1), establishing a network connecting each computer system in the plurality of computer systems with a server system (See Gafken Col. 3 Paragraph 6 and Col. 12 Paragraph 7); generating an update partition file within the server system (See Gafken Col. 12 Paragraph 7 – Col. 13 Paragraph 1); the update partition file includes an entry (See Gafken Col. 12 Lines 53-67 "code image") and a plurality of encrypted elements (See Gafken Col. 12 Lines 53-67 "BIOS Signature" wherein one interpretation of the claims each binary bit of the signature reads on one of the plurality of encrypted elements), each encrypted element within the plurality of encrypted elements is associated with the entry (See Gafken Col. 12 Line 53 - Col. 13 Line 2), transmitting the update partition file over the network to each computer system in the plurality of computer systems (See Gafken Col. 12 Paragraph 7); and following determining that the update partition file is stored within the computing system for updating the protected partition, verifying whether the entry within the update partition file has been generated by the server system (See Gafken Col. 12 Line 53 – Col. 13 Line 2); and storing the update partition file within the predetermined location of each computer system in the plurality of computer systems (See Gafken Col. 12 Paragraph 5), wherein the entry within the update partition file is written to the protected partition only following verification that the entry has been generated by the server system (See Gafken Col. 12 Line 53 – Col. 13 Line 2).

However, Gafken failed to disclose the protected partition being within a hard drive; or that a setup password stored in the nonvolatile storage for use in the predetermined security procedures; wherein the initialization routine includes instructions causing the processor of the computer system to perform a method including: comparing information stored in the protected partition with information from the update partition file stored within the predetermined location; or that when a portion of the information stored in the protected partition is found partition is found to match a portion of the information stored within the update partition file, overwriting the portion of the information stored in the protected partition with the portion of the information stored in the update partition file if space around the portion of the information stored in the protected partition is sufficient; or that when a portion of the information stored in the protected partition is not found to match a portion of the information stored within the update partition file, writing the portion of the information stored within the update partition file to append to the information stored in the protected partition if space within the protected partition is sufficient; or that the update partition file included a plurality of entries and that each entry within the plurality of entries includes information to be stored at a different location within the protected partition; or verifying whether each entry in the plurality of entries within the update partition file has been generated by the server system; or that each entry in the plurality of entries within the update partition file is written to the protected partition only following verification that the entry has been generated by the server system. Gafken further failed to disclose a plurality of encrypted elements (in a more specific interpretation of encrypted element than a bit of a signature).

Gafken did disclosed that "although the example…describes a flash memory used to store…a BIOS…other types of nonvolatile memories storing other types of information may be used" (See Gafken Col. 14 Paragraph 6).

Arnold teaches that a BIOS can be stored in a protected partition of a hard drive (See Arnold Col. 2 Line 63 – Col. 3 Line 12).

It would have been obvious to the ordinary person skilled in the art at the time of invention to employ the teachings of Arnold in the BIOS updating system of Gafken by storing the BIOS in a protected partition of a hard drive instead of flash memory. This would have been obvious because the ordinary person skilled in the art would have been motivated to provide a fast and efficient way to store BIOS code.

Menezes teaches that providing a sequence number (password), stored and updated at both a receiver and a sender, in a digital signature of the sender, protects the signature against replay attacks (See Menezes Page 399 Section (ii).

It would have been obvious to the ordinary person skilled in the art at the time of invention to employ the teachings of Menezes to the validation signatures of Gafken by providing a sequence number in the signature of the update image. This would have been obvious because the ordinary person skilled in the art would have been motivated to provide protection against illicitly signed updates.

Hasbun teaches that a BIOS update can include a plurality of entries (blocks) wherein each entry within the plurality of entries includes information to be stored at a different location within the BIOS (See Hasbun Col. 5 Paragraph 6- Col. 6 Paragraph 2); comparing information stored in the protected partition with information from the update partition file stored within the predetermined location and when a portion of the information stored in the protected partition is found partition is found to match a portion of the information stored within the update partition file, overwriting the portion of the information stored in the protected partition with the portion of the information stored in the update partition file if space around the portion of the information stored in the protected partition is sufficient (See Hasbun Col. 5 Paragraph 6 – Col. 6 Paragraph 2 and Col. 12 Line 59 – Col. 16 Line 27 wherein Hasbun teaches that a bios update can be allocated into virtual blocks so that the blocks can be updated individually without having to erase the entire memory first); when a portion of the information stored in the protected partition is not found to match a portion of the information stored within the update partition file, writing the portion of the information stored within the update partition file to append to the information stored in the protected partition if space within the protected partition is sufficient (See Hasbun Col. 7 Paragraph 2 wherein Hasbun teaches that new blocks should be allocated from existing free memory).

It would have been obvious to the ordinary person skilled in the art at the time of invention to employ the teachings of Hasbun to the bios updating system of Gafken, Arnold, and Menezes by updating each update block of the BIOS one at a time in the manner taught by Hasbun. This would have been obvious because the ordinary person skilled in the art would have been motivated to provide a safe method for updating a bios without risking loss of the entire bios in the event of a power failure.

Hayashi teaches a method for providing a variety of software safely by breaking the file into pieces and decrypting each piece separately (See Hayashi Page 1 Col. 2 Paragraphs 3-10).

It would have been obvious to the ordinary person skilled in the art at the time of invention to employ the teachings of Hayashi to the updating system of the combination of Gafken, Arnold, Menezes, and Hasbun by signing blocks of the file separately from the other blocks, thereby obtaining a signature (encrypted element) for each block (entry), and thereby allowing each signature to be verified separate from the other blocks. This would have been obvious because the ordinary person skilled in the art would have been motivated to provide users with customized software without imposing too much of a load on the provider.

Regarding claims 45 and 58, the combination of Gafken, Arnold, Menezes, Hasbun, and Hayashi disclosed forming a first message digest by applying a hash algorithm to said entry, and forming a second message digest by signing said encrypted element associated with said entry using a public key of said trusted server system, and determining that said first and second message digests are identical (See Gafken Col. 12 Paragraph 7 Line 10 – Col. 13 Line 2).

Regarding claims 48 and 61, the combination of Gafken, Arnold, Menezes, Hasbun, and Hayashi disclosed that information stored in said protected partition is compared to each entry in said plurality of entries within said update partition, when a matching portion of said information stored in said protected partition is found to be similar to said entry, said matching portion is overwritten with said entry if space around said matching portion is sufficient, and when a matching portion of said information stored in said protected partition is not found to be similar to said entry, said entry is appended to said information stored in said protected partition if space within said protected partition is sufficient (See the rejection of claim 38 above).

Regarding claims 46 and 59, the combination of Gafken, Arnold, Menezes, Hasbun, and Hayashi disclosed the predetermined setup procedures include verifying that said update partition file has been generated by said trusted server system includes signing an encrypted portion of said update partition file with a public key of said trusted server system, and said encrypted portion of said update partition file has been prepared by signing, with a private key of said trusted server system, a result of the application of an algorithm to data including a version of said setup password accessed by said trusted server system (See the rejection of claim 37 above and Col. 12 Paragraph 7 – Col. 13 Paragraph 1).

Regarding claims 47 and 60, the combination of Gafken, Arnold, Menezes, Hasbun and Hayashi disclosed that the data includes said version of said setup password appended to a portion of said update partition file, said algorithm is a hash algorithm generating a message digest (See Gafken Col. 12 Paragraph 7 – Col. 13 Paragraph 1), and verifying that said update partition file has been generated by said trusted server system includes applying said hash algorithm to said setup password stored within said computing system appended to a portion of said update partition file to generate a first version of a message digest and comparing said first version of said message digest with a second version of said message digest obtained by signing said encrypted portion of said update partition file (See Gafken Col. 12 Paragraph 7 – Col. 13 Paragraph 1).

Claims 49 and 62 are rejected under 35 U.S.C. 103(a) as being unpatentable over the combination of Gafken, Arnold, Menezes, Hasbun, and Hayashi as applied to claim 48 above, and further in view of Schmidt (U.S. Patent Number 5,826,015).

The combination of Gafken, Arnold, Menezes, Hasbun, and Hayashi disclosed a secure bios updating system (See rejection of claim 38 above) but failed to disclose requiring a user to input a password to unlock the bios write capabilities. However, Gafken, Arnold, Menezes and Hasbun did disclose the use of password challenges (See Gafken Col. 12 Paragraph 7 – Col. 13 Paragraph 1).

Schmidt teaches that in order to remotely upgrade a bios, an administrator password should be provided in order to unlock the partition (See Schmidt Fig. 9 and abstract).

It would have been obvious to the ordinary person skilled in the art at the time of invention to employ the teachings of Schmidt to the bios updating system of Gafken, Arnold, Menezes, Hasbun, and Hayashi by requiring a correct password to be entered in order to unlock the bios altering capabilities. This would have been obvious because the ordinary person skilled in the art would have been motivated to protect the current bios from accidental or illicit alterations.

(10) Response to Argument

Issue #1

The appellant argues that the combination of references relied upon by the examiner does not teach or suggest that the update partition file includes a plurality of entries and a plurality of encrypted elements, wherein each encrypted element in the plurality of encrypted elements is associated with an entry in the plurality of entries, wherein the method performed by the initialization program includes verifying whether each entry in the plurality of entries has been generated by the server system, and wherein each entry in the plurality of entries is written to the protected partition only following verification that the entry has been generated by the server system.

The examiner presents that there are two differing interpretations of the claim language in which the combination of references reads on these claim limitations. Both of these interpretations have been discussed below.

Regarding the limitation that "the update partition file includes a plurality of entries and a plurality of encrypted elements", the examiner presents that the combination of references does teach this limitation. Gafken, in Col. 12 Line 53 - Col. 13 Line 2, disclosed a code image with a code image signature for updating a BIOS, which is equivalent to the update partition file of the claim language, wherein the code image is equivalent to "an entry" and the code image signature is equivalent to "the plurality of encrypted elements".

With regards to the plurality of entries, Hasbun renders obvious that in a BIOS update, the update data can be divided into blocks. These blocks read on the plurality of entries.

With regards to the plurality of encrypted elements, the code image signature by itself can be seen to fall within the scope of this claim element. First, it is important to note that a digital signature is generated by computing a digest and encrypting the digest with the private key of the signor. Then a verifier can verify the signature by decrypting the signature with the public key of the signor, and comparing the decrypted digest with a digest created in the same manner as the digest which was encrypted. Second, it is important to note that a digest is created by a function that takes an input string, which varies in length or bits, and produces a fixed length, generally smaller, output string. In the case of Gafken, the input string is the code image, and the output string is the digest value, which is then encrypted, thereby becoming the code image signature. This signature contains multiple bits, each bit reading on one of the plurality of encrypted elements.

Furthermore, the teachings of Hayashi suggest that in software updating, blocks of the update data should be encrypted separately in order to relieve the burden on the update servers. As such, one of ordinary skill in the art would have found it obvious to sign (calculate a digest and encrypt) each block of the code image of Gafken and Hasbun separately, thereby creating an alternative plurality of encrypted elements.

As such, it can be seen that the limitation of the update partition file including a plurality of entries and a plurality of encrypted elements has been met by the combination of prior art references.

Regarding the limitation that "wherein each encrypted element in the plurality of encrypted elements is associated with an entry in the plurality of entries", the examiner believes that the combination of references relied upon meets this limitation. In the first scenario, wherein the signature is calculated from the entire code image, each bit in the signature is associated with each and every entry of the plurality of entries (blocks) because there is a single signature for all of the blocks. In the second scenario, wherein a separate signature is calculated for each block, each signature is associated with the block (entry) from which it was calculated. As such, the examiner believes that this limitation is met by the combination of references.

Regarding the limitation that "wherein the method performed by the initialization program includes verifying whether each entry in the plurality of entries has been generated by the server system", the examiner believes that this limitation is met by the combination of applied references. Gafken clearly disclosed verifying that the code image was generated by the server system, as can be seen in Gafken Col. 12 Line 53 - Col. 13 Line 2. In the first scenario, wherein the code image was divided into blocks and a signature was calculated over the entire code image, in order to verify the signature, the blocks would have to be recombined to form a single code image again so that the correct digest could be calculated for comparison with the code image signature. Therefore, all of the blocks (entries) would be verified. In the second scenario, wherein a separate signature was calculated for each block (entry) of the code image, in order to verify the [whole] code image as disclosed by Gafken, the signature of each block would need to be verified. As such, the examiner believes that combination of applied references meets this limitation.

Regarding the limitation wherein each entry in the plurality of entries is written to the protected partition only following verification that the entry has been generated by the server system, the examiner believes that this limitation is met by the combination of applied references. Gafken, in Col. 13 Lines 4-6, clearly disclosed that the update process only continues if the code image is validated. As can be seen in Fig. 5 of Gafken, the validation step 505, or 560, is performed prior to updating (storing) the code. According to the disclosure of Gafken, if the code image is not validated, the code is not updated. In the first scenario, wherein there is as single signature over all of the code image blocks, all of the blocks must be validated for any of the blocks to be stored. Therefore, if any block is invalid, the whole code image would be invalid, and thus the code would not be updated. In the second scenario, wherein there is a separate signature for each block, two different situations would both be obvious, and both would read on the claim language. The first is that, because Gafken teaches that the [whole] code image must be validated for the code to be updated, each signature would be validated prior to updating the code. The second is that, because each signature can be validated individually, each block of code could be updated after the block was validated. Based on the disclosure of Gafken, it would not be obvious to update a block of code which either had not been validated or was determined to be invalid. Therefore, the examiner believes that the combination of references reads on the claim limitation.

Because the examiner has shown that the combination of references reads on the contested claim limitations, the examiner believes that the rejection should be upheld.

Moreover, the appellant appears to believe that the limitation "wherein each entry in the plurality of entries is written to the protected partition only following verification that the entry has been generated by the server system", requires that each entry is verified separately. The examiner points out that, although separately verifying each entry falls within the scope of this limitation, the limitation is much more broad. As shown above, a single verification of all entries via a single signature falls within the scope of this claim as well, because an entry is updated only when all entries have been verified, and therefore only updated when each entry has been verified, via verification of the single signature. This does not require separate verification of each entry, but rather verification of all entries is performed at once, while still falling within the scope of the claim language. Similarly, the claim does not require separate verification of each entry, but rather requires that no entry is stored in the protected partition unless it has been verified. What the appellant believes the claim language states, and what the claim language actually states are quite different. However, as discussed above, the examiner has shown it to be obvious to conduct separate verification of each block of the code image, and as such the examiner believes that in either interpretation the rejection should be upheld.

Further, the examiner points out that the claim language does not recite that "every verified entry is stored to the protected partition, while every entry not verified is not stored to the protected partition", as the appellant appears to argue in the brief on page 16 Lines 1-4. Rather, the claims simply recite that an entry is not stored prior to it being verified, which Gafken clearly disclosed not updating the BIOS data unless the code image was validated.

Furthermore, the appellant's arguments on page 15 Lines 3-19 of the brief recite alleged advantages to separate verification of entries. Because neither separate verification of entries, nor any alleged advantages of separate verification have been claimed, the examiner has not further addressed these arguments.

(11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

/MATTHEW HENNING/

Matthew Henning
Patent Examiner
Art Unit 2131

KIM VU
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100

Conferees:

Kim Vu KV
Supervisory Patent Examiner
Art Unit 2135

Hosuk Song HS
Primary Patent Examiner
Art Unit 2135

HOSUK SONG
PRIMARY EXAMINER